

Date - last edited: 2021-11-01

A basic guide for using VIM

(From commando-mode, enter ':help' for the Vim manual).

---

## 1. Working in last-line, insert, and commando-mode.

### 1.1 - Differences between the three modes:

In insert-mode, it is possible to write/delete normal characters to the document. To edit it in a more general sense. It is the 'standard' mode for writing a document in Vim.

In commando-mode, it is possible to input various commands. This includes functionality for deleting/inserting whole lines, moving between various sections, copying text, and various other things.

In last-line mode, one enters final commands. For example to quit vim, to save the file, and similar things.

Vim also has other modes. For example the visual mode.

### 1.2 - Switching between the three modes:

When opening up Vim, it starts in commando-mode.

To enter insert-mode, enter 'i'.

To re-enter commando-mode, hit 'esc'.

To enter last-line-mode, enter ':' from commando-mode.

---

## 2. Starting and exiting the Vim-texteditor.

### 2.1 - Start Vim:

Open a shell, and enter 'vim'. One can add various commands when starting it, so that for example a file is opened at the same time. To open a file at the same time, simply use 'vim pathToFile'.

### 2.2 - Exiting Vim:

Enter command-mode:

':q' - quit.

`':q!'` - quit without save.

`':w'` - save.

`'ZZ'` - save and quit(enter from command-mode).

---

### 3. Creating new text-files, and opening them from inside Vim

#### 3.1 - Creating New Files:

When in commando-mode, enter `':e newFile'`, and this will create a new file in the default text-standard.

#### 3.2 - Opening Files:

`':e fileName'` - opens a file named 'fileName', if in cwd.

`':Ex'` - browse for the file from the current cwd.

`':Ex path'` - browse for the file from the stated directory.

---

### 4. Working with text.

#### 4.1 - Entering text:

To enter text, one only need to enter the insert-mode. After having done so, one can use the keyboard as one normally would. There are no, it seems, special tricks. The special tricks are found in commando-mode.

#### 4.2 - Editing Text:

`'x'` - deletes the marked character.

`'u'` - undo the latest event.

`'ctrl+r'` - undo the latest undo.

`'dw'` - delete a whole word. Must be at beginning of word.

`'d0'` - delete to the beginning of the line.

`'d$'` - delete to the end of the line.

`'dgg'` - delete to the beggining of the file.

`'dG'` - delete to the end of the file.

---

### 5. Traversing the text.

## 5.1 - Basic Commands:

'h' - left one character.  
'j' - down one line.  
'k' - up one line.  
'l' - right one character.  
'0' - beginning of the line.  
('\$' - end of the line.  
'w' - forward one word.  
'b' - backward one word.  
'G' - end of the file.  
'gg' - beginning of the file.  
'\.' - last edit.

## 5.2 - General Tips:

Set a number prior to the command, and the intended command will be executed several times.

-----

## 6. - Highlight, cut, copy, and paste text

### 6.1 - Highlight:

'v' - highlight text for editing.  
'V' - highlight line/lines for editing.  
'ctrl+v' - highlight whole column.

### 6.2 - Cut:

Highlight text, and then enter 'd'.  
This will move the text into the copy-buffer.

### 6.3 - Copy:

Highlight text, and then enter 'y'.  
This will move the text into the copy-buffer.

### 6.4 - Paste:

Enter 'p' from command-mode, and the text in the copy-buffer will be pasted after the current line. 'P' will paste on the current line.

-----

## 7. Search and replace text.

### 7.1 Search:

'/test' - Search for the string 'test' in file.

### 7.2 Replace:

'%s/test/newtext/g' - Search for the string 'test' in whole file, and replace this string with 'newtext'. Add 'c' to the 'g', to confirm replacing.

-----

8. Vim Alternatives(Vim no longer community project).

NeoVim

9. Sources.

<https://www.linux.com/learn/vim-101-beginners-guide-vim>